REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

SPEECH SYNTHESIS USING LONG-TERM SHORT MEMORY AND RECURRENT NEURAL NETWORK (LTSM-RNN)

ARKAN ADNAN IMRAN AL-YASARI

Master's Thesis

Supervisor

Prof. Dr. GALİP CANSEVER

Istanbul , 2022

SPEECH SYNTHESIS USING LONG-TERM SHORT MEMORY AND RECURRENT NEURAL NETWORK (LTSM-RNN)

Arkan Adnan Imran AL-YASARI

Electrical and Computer Engineering

Master's thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titied SPEECH SYNTHESIS USING LONG-TERM SHORT MEMORY AND RECURRENT NEURAL NETWORK (LTSM-RNN) prepared by ARKAN ADNAN IMRAN AL-YASARI and submitted on 17/12/2022 has been **accepted unanimously** for the degree Master of Science in Electrical and Computer Engineering

Prof. Dr. GALİP CANSEVER

Supervisor

Thesis Defense Jury Members:

Prof. Dr. GALİP CANSEVER	Department of Engineering and Architecture, Altinbas University	_____
Prof. Dr. Osman Nuri UCAN	Department of Engineering and Architecture, Altinbas University	_____
Asst. Prof. Dr. Tariq Abed MUHAMMAD	Department of Computer Science, Imam Jaafar Al-sadiq University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission data of the thesis to Institute of Graduate Studies: ____/____/____

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Arkan Adnan Imran AL-YASARI

Signature

DEDICATION

I devote and pledge this research work to my supervisor who is salient for guiding me through whole research work as well as my family for always assisting me in my hard time.

PREFACE

First and foremost, I would like to thank my supervisor Prof. Dr. GALİP CANSEVER for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Prof. Dr. GALİP CANSEVER a couple of times every week helped me tremendously in understanding the logic behind the research. It made me better realize the technical need for this research work.

ABSTRACT

SPEECH SYNTHESIS USING LONG-TERM SHORT MEMORY AND RECURRENT NEURAL NETWORK (LTSM-RNN)

Al-Yasari, Arkan Adnan Imran,

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor. Prof. Dr. Galip Cansever

Date: / 2022

Pages: 46

Over the past decade, artificial intelligence has made lifelike computer-generated multimedia content like images, audio, and video widely available. AI-generated information can be used in legal and illegal ways, for as evidence in a court case. Automatically distinguishing AI-generated speech from natural speech is becoming increasingly important. We studied manipulable audio, specifically voice. Our goal is to enhance automatic sound recognition by employing bi-spectral analysis and a Long-Term-Short Memory Recurrent Neural Network (LTSM-RNN). Our voice dataset is rich in natural and synthetic sounds produced by various methods. We classified all speech recordings using support vector machines (SVM), logistic regression (LR), and convolutional neural networks (CNN) to extract bicoherence (binary classifications among real as well as false voices and multi-label classifications that distinguish one type of voices from the others). After estimating bicoherences from audio recordings, the experiments were done. We first tried to categorize the bicoherences using simple multiclass and binary classifications with an LTSM, many RNN, and some CNN, repeating earlier work by mean, standard deviation, skewness, and kurtosis extraction across modules and stages. Next, we used LTSM to mimic an open set scenario where the model was evaluated with data it had not seen during training. We categorize the newly extracted features using basic multi-label and binary classifications utilizing hybrid LTSM-RNNs. This method yielded the greatest results with 99.76% accuracy. The research combined the two lists of features for additional

categorization (in this case, also using an open set environment). The findings helped explain how bi-spectral analysis distinguishes between authentic and faked speech recordings and encouraged multimedia forensics research.

Keywords: Forensic, Artificial Intelligence, Speech Synthesis, Kurtosis, Classification.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATION

LR	:	Logistic Regression
SVM	:	Support Vector Machine
RF	:	Random Forest
LPCCs	:	Linear Prediction Cepstrum Coefficients
TTS		Text To Speech
VC	:	Voice Conversion
MFCCs	:	Mel-Frequency Cepstrum Co-Efficient
CNN	:	Convaional Neural Network
DFT	:	Discrete Fourier Transform
RNN	:	Recurrent Neural Network
MK	:	Mean-Kurtosis

1. INTRODUCTION

1.1. SPEECH SYNTHESIS OVERVIEW

Interacting with multimedia data generated by artificial intelligence is a growing trend in the present day. Twitter, Facebook, Instagram, WhatsApp, and Telegram are among the numerous social media platforms that have facilitated the dissemination of information and multimedia content such as images, movies, and audio recordings to nearly anybody on the planet. False information spreading is one of the most frequent unintentional effects of social media use.

Now, though, AI applications are becoming more accessible and widespread. Synthesized speech and other audio signals are prevalent in contemporary society. Synthetic voices are utilized by audiobooks, navigation systems, mobile phones, tablets, and voice assistants like Google Home and Amazon Alexa. Numerous contact centers, hospitals (where speech synthesis helps patients with phonatory abnormalities, dyslexia/dysgraphia, or visual impairments), and public transit systems also employ this technology.

Several apps that can alter or produce speech using an algorithm have become more accessible and popular. Even while these devices can be utilized in a variety of exciting ways, in the wrong hands they could be hazardous. The widespread usage of information generated by artificial intelligence (AI) and the ease with which individuals can transmit it raises questions about how much of it is legitimate and how much is fraudulent.

Given the fragility of human speech, it is conceivable that select victims of wiretapping could have their conversations electronically altered in order to be used as courtroom evidence. As a result of the altered auditory evidence, errors in judgment are possible [1]. It is also possible to force a public figure (such as a notable politician) to say something they would never say in their normal voice. A faked signal could be utilized to alter diplomatic relations between nations or election outcomes. [2]. Multimedia forensics is one application of this research, and the preceding examples illustrate how crucial it is to have a solid grip on the construction of audio recordings, especially those incorporating speech (and more specifically audio forensics).

Our research aims to develop internationally applicable automatic systems that can distinguish between speech synthesis based on artificial intelligence and actual speech

recordings.

Bi-spectral analysis was employed for this purpose; this analysis computes the bi-coherence, a complicated characteristic generated from the temporal frequency representation of an audio sample. We hypothesize in this research that created speech recordings will exhibit higher order correlations than actual voice recordings. Nonlinear processing of the signals is responsible for these correlations; it modifies the spectral content of the signals, making it more difficult to identify them using shared criteria such as power spectrum. Since bi-coherence is a 3rd-order property, as opposed to the power spectrum, which is a second-order property, it follows that the bi-coherences estimated from the fake voice recordings and those of Bonafede would be distinct.

After calculating the bicoherences, they were subjected to a variety of machine learning algorithms to determine how to classify them. We were able to produce three separate collections of features using bi coherences: recurrent neural network (RNN) features, mean-kurtosis RNN features, and MKU features. For Bi-coherence modules and phases, the mean, standard deviation, skewness, and kurtosis (MK characteristics) are determined. In [3] a method for extracting these traits is proposed. Five distinct RNNs were trained using the modules of the recovered bi-coherences from the aforementioned five classes of simulated speech in order to derive RNN features. After training, we equipped five distinct RNNs with all of the modules from the various classes of bi-coherences, resulting in five distinct image reconstructions for each speech sample. MSEs between the "compressed" and reconstructed versions of a picture are associated with the RNN's characteristics (obtained through the encoder component of the RNN). This method generates five distinct MSEs for each audio file including human speech. The final phase entails combining the MK and RNN features to form the MKU features. We categorized binary (authentic vs. false) and multiclass MK, RNNsupport vector machine (SVM), logistic regression (LR), as well as a random forest to analyze the relationship between these variables and the MKU features (RF).

Several support vector machines were used to create our open set environment. Under these conditions, each SVM, with or without the aid of known unknown features, does a one-versus-the-rest classification. We hope that the results will encourage more research into audio (and speech) forensics and shed light on how bi-spectral analysis may discriminate between natural and fake speech.

1.2. RESEARCH MOTIVATION

In this study, there will be a brief discussion of techniques for lowering the number of dimensions, such as feature extraction and normalization. The classification problem will be thoroughly studied shortly after the debut of machine learning. When choosing how to classify our features, we will also investigate several additional classification algorithms such as logistic regression, random forests, and support vector machines. In the end, you'll see some results of our efforts using convolutional neural networks as well as convolutional autoencoders. We will show and analyze several measurements, such as confusion matrices, accuracy, as well as receiver operating characteristics (ROC) curves, as we develop open set classification to evaluate our classifiers.

In this article, we'll explore the idea of features extracted from various types of multimedia content. The many methods of post-processing, such as feature normalization, will also be discussed. Both mean-kurtosis RNN (MKU) and mean-kurtosis (MK) features are used to measure the dispersion of a distribution [4], [5]. recovered using bicoherences, will be given in 2D and 3D, respectively, using dimensionality reduction approaches that will be thoroughly discussed. Features extraction often refers to a set of signal compression techniques that preserve pertinent information while decreasing the amount of data a digital system must process. By analyzing a signal constituted of minute-by-minute data, one can estimate, for instance, the daily mean and variance of a specific room's temperature. This method allows us to maintain vital data, such as temperature, while substantially reducing the amount of data we must keep.

In audio processing, the spectrogram, the Mel-Frequency Cestrum Coefficient, and the linear Prediction Cestrum Coefficients are the most utilized properties (LPCCs).

These features are typically extracted from the time series audio stream y(k) via segmentation and windowing methods[6], [7]. The most widely used window functions in audio processing are the rectangular, Hamming, and Hann windows. Each picture slice is divided by the window function when windowing an image. The term "segmentation" refers to the process of dividing a signal in smaller parts so that they can be examined

independently.

The discrete Fourier transform (DFT), also called as the quick Fourier transform, is commonly employed as the first step in feature extraction approaches since it can be applied to audio signals and gives a frequency representation of the data (FFT). In multi-media processing, feature extraction is a crucial step[8], [9]. Researchers in computer vision still struggle to extract features from images and audio that correctly represent the underlying structure [10]–[12]. Changes may be made to the alleged value range of reclaimed characteristics after they have been reclaimed. We will demonstrate two of the most well-known methods for leveling your facial traits.

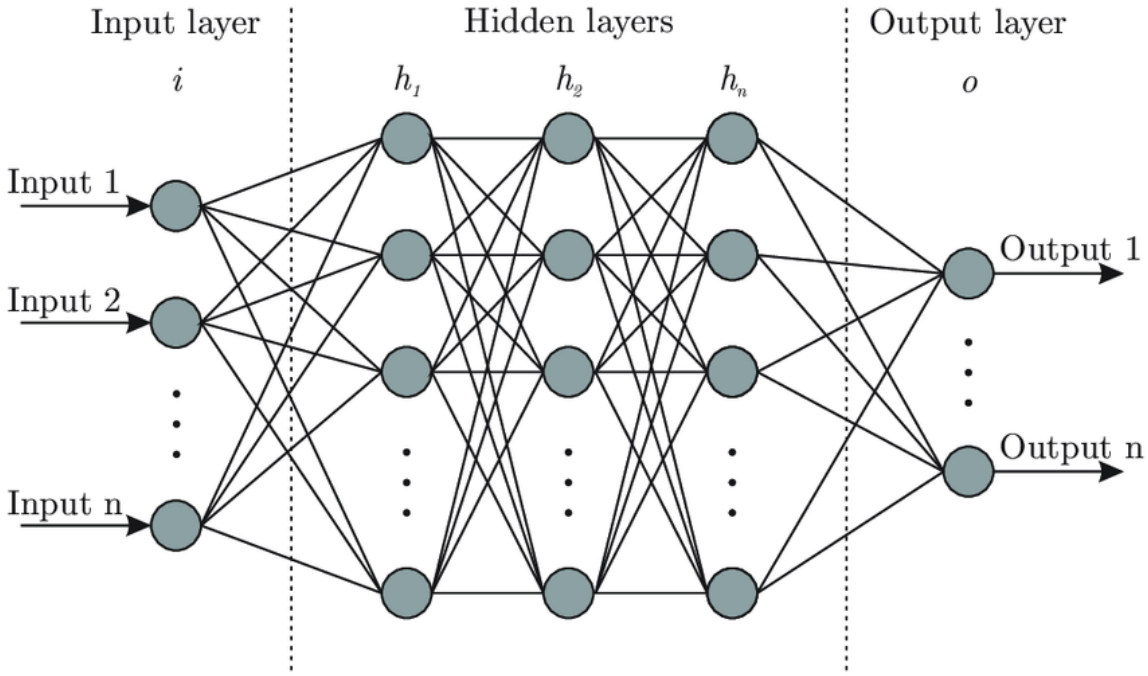

Figure 1.1: Design neural network configuration for categorization.

The encoder and decoder are, respectively, two components of a recurrent neural network called a recurrent auto-encoder. The encoder component of the network receives the original image and outputs a compressed version, whereas the decoder component takes the encoder's compressed output and outputs a nearly identical reconstructed image (See Figure 1.2).

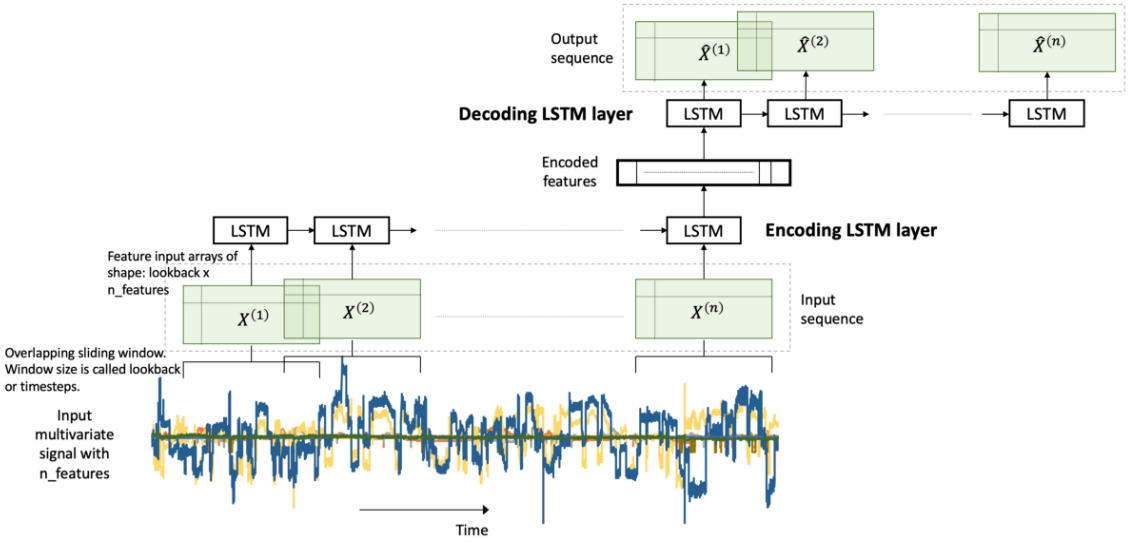

Figure 1.2: LTSM-RNN auto-encoder structure.

The primary benefit of RNNs is their greater capacity to recreate the class of images utilized during training. An auto-encoder will provide a better reconstruction, for instance, if it was trained on photos of dogs rather than other animals[13], [14]. Auto-encoder is frequently used in the extraction of image characteristics, detection of outliers, and separation of image objects. The author contrasts PCA with robust Auto-encoders, which are utilized for anomaly identification, in [15], [16]. In contrast to a normal auto-encoder, an RNN connects its deeper layers to its shallower layers via skip connection (or concatenation) layers.

1.3. AIM OF STUDY

We have illustrated the open-set classification answer with this demonstration. An open set setup requires a model to differentiate between "known" classes used to train the model and "unknown" classes not seen during training.

Following the presentations, we classify an open set using the following three primary data sources:

i. The LTSM-RNN model must reliably detect and categorize "Known Data," which consists of training and testing data from known classes.

ii. "Known Unknown Data" (KUD) refers to data that is known at training time but is considered unknown to simulate unknown data at training time when utilizing the LTSM-RNN approach for both training and testing.

iii. In the LTSM-RNN methodology, "unknown unknown data" refers to information used exclusively for testing the model with cases from classes that have never been observed.

Classification yields two values for each test set input xt: the actual class to which it belongs and the expected class to which it has been assigned. To better illustrate the efficacy of the categorization algorithm, we may collect this information into a table. Each row of the table contains the projected class instances, whereas the columns contain the actual class instances. For a representation of the 2x2 matrix relevant to two-class problems,

1.4. PROBLEM STATEMENT AND FORMULATION

We consider a dataset consisting of many digital audio signals serving as speech recordings to shed light on the matter. These noises were produced using either a speech synthesis program or a recording from actual human voices (spoofed or fake speech). We assume that a voice recognition program was used to create the synthetic noises, maybe a combination of text-to-speech (TTS) and voice conversion (VC). The fundamental objective of this study is to design an automatic method that can take as input either authentic or spoofed voice signals and accurately determine which is which. This can be viewed as a categorization dilemma between two groups. To tackle this issue, we will employ bi-spectral analysis, which, as we saw in Chapter 2, has the advantageous virtue of remembering the higher-order correlations that nonlinear processing adds to speech signals. Nevertheless, bi-spectral analysis is a potent tool, and it might be used to discover more about the voice signals than whether they are legitimate. As a result, we can broaden our focus and establish three primary goals for this thesis.

i. Using LTSM-RNN, it is feasible to distinguish between authentic and fabricated voice recordings.

ii. Categorization of the LTSM-RNN algorithm used to generate the parody speech.

iii. Using the LTSM-RNN approach, forged speech generated by previously unknown voice synthesis algorithms can be recognized.

As stated previously, the first objective may be achieved with only two elementary classes, whereas the second requires a more complicated set of classes.

1.5. ORGANIZATION OF THESIS

To facilitate comprehension, we have structured our work as follows: in Chapter 1, we discuss feature extraction and normalization, dimensionality reduction techniques, and classification methods. The second chapter examines voice synthesis techniques, multimedia forensics, and ways presented in the literature for distinguishing between naturally spoken language and language manufactured by computers and artificial intelligence. In Chapter 3, our method will be presented in great depth; in Chapter 4, we will demonstrate how we classified open and closed sets using the MK, RNN, and MKU features. Using the LTSM-RNN technique, we will put our models to the test using a huge and diverse dataset of voice samples (both actual and synthetic speech signals). Chapter 5 will give the research findings and their analysis in conclusion.

2. RELATED WORK

In this chapter, we'll look at Text-to-Speech (TTS) and Speech-Conversion voice synthesis algorithms after a quick introduction to multimedia forensics methods (VC). We will also go through some of the current research that has been done to distinguish between natural and synthetic speech, such as data-driven and classical feature-based approaches. While bi-coherence and spectrogram are used for feature extraction and as inputs to classifiers in the first two approaches, data-driven features are also used in the third.

Multimedia forensics investigates suspects using audio, video, and image analysis. This metadata may identify the acquisition equipment responsible for the content, the signal's integrity,[17] the environment in which it was collected, digital signal processing, or synthetic versus natural material. Multimedia forensics assumes that recording, encoding (using compression), and editing leave traces that can be recovered by specialist analysis.

Active and passive methods can verify multimedia content. Active techniques often use "watermarks" or "digital signatures" left by equipment upon acquisition before compressing and saving content to a hard disk (digital camera or microphone). Passive techniques assume that all digital operations on a signal leave trace and merely require the application of the content, as previously indicated. Multimedia forensics is mostly passive [18]–[22].

Photographs and films are marked by lenses, camera sensors (typically CCD or CMOS), and color filter arrays [23]. Finally, the color filter creates other aberrations, such as lateral chromatic aberration, that can be used to identify the camera that took the photo or reveal image changes [24]–[26]. Sensor pattern noise, primarily camera-specific photo response nonuniformity (PRNU), can be utilized to identify images' sources [27]–[29].

JPEG compression, which loses information irreversibly, can be used to detect digital forgeries [30], [31]. After many trials, picture alteration has been determined (sometimes called "post-processing"). This can be done by examining the scene's shadows, lights, reflections, perspective, and object shape [32], [33] or the signal (to check for signal artifacts caused by the adjustments) (to check for artifacts on the signal produced by the alterations).

Audio forensics can determine the source microphone, time, place, and signal kind. These methods use audio signal capture traces. Checking for compression helps determine an audio stream's bitrate. Signal processing processes like applying nonlinear effects or combining audio signals and post-production editing operations like adding or removing audio data (cut-type edits) can also be determined [34]. Mel-Frequency II. Cestrum The coefficient is the

differentiating qualities of audio recordings categorized by microphone. (MFCCs).
Discontinuities in the temporal domain [35], [36], shifts in audio power in the following frame (fuzzy clustering), substantial variations cut-type editing on an audio signal (E-net frequency) can be detected using properties of lossy audio compression [37] and short-time energy in the background noise region (E-net frequency). Least squares provide a means of determining mixing parameters [38].
Bi-spectral analysis can identify fake speech by exhibiting nonlinear processing on audio data. Bi-spectral analysis can reveal higher-order speech signal correlations since bi-coherence is a third-order characteristic. Thus, bicoherences from authentic signals differ from those from false voice recordings. We can develop AI and ML algorithms to detect these distinctions [3], [39]. We will show how to detect digital forgeries in voice signals in the next chapters. We will also illustrate our strategy on a large and diverse voice data set.

One of the most convenient and unobtrusive methods of biometric identification is voice recognition. ASVs are one type of voice-recognition technology. Using a technique known as a "presentation assault" [40], it is possible to trick any biometric system, even an ASV system. Speech synthesis, replay, and imitation are regularly employed among the many techniques used to undermine a presentation.

An imposter will change their tone of speech to mimic the target. To "replay" a user's voice is to play back an audio recording of the user's voice and show it on the ASV software system in place of the user's actual voice at the time of recording. The techniques of speech synthesis, such as making a computer-generated voice seem more human, will take up the bulk of this article. Popular speech synthesis applications include voice converters and text-to-speech (TTS) programs (VC). When given some text, text-to-speech technology generates an artificial voice that "reads the text aloud," while voice conversion technology takes one person's voice and modifies it so that it sounds like it was created by someone else.

According to [41], a text-to-speech (TTS) synthesizer is comprised of two primary components. A natural language processing (NLP) component that can generate a phonetic transcription of the original text forms the basis of the system. In order to generate synthetic speech with a human-like voice, the symbolic data from the first module is processed by a digital signal processing unit in conjunction with information on the desired intonation and rhythm. Now that we've established that, let's examine each separately.

The NLP component is made up of the "natural" prosody generator, the morph syntactic analyzer (MSA), and the letter-to-sound converter (LTS) (PG).

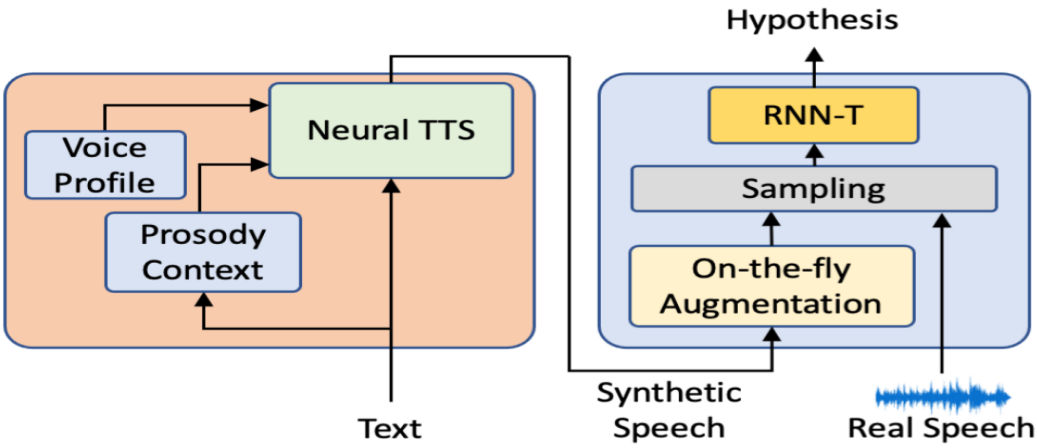

Figure 2.1: Design principles for a TTS simulated voice for teaching speech recognition software.

A digital signal processing module generates the audio waveform for the spoken signal, while an NLP module provides data on the phonemes and prosody of the speaker's language. The model of the vocal tract is based on three traditional synthesis techniques: There are a few different types of synthesis that can be used to create synthetic speech, including syllable synthesis, log - likelihood synthesis, as well as articulation synthesis. Independent format filters are used in formant synthesis to accurately predict the vocal tract's transfer function. While a noise signal is used to generate unwanted sounds, a stream of impulses is used to generate speech. Despite the difficulties inherent in altering the settings of the synthesizer, the parameters of the formant synthesizer can be determined based on a set of criteria gleaned through studying mobile phone usage and its surrounding environment. Like syllable synthesis is the linear prediction technique, with the primary distinction being that an all-pole filter is used to generate all sounds in the linear prediction method, whereas parallel filters are typically utilized in formant synthesis. To estimate the characteristics of the vocal tract, linear prediction employs the available data. The concept of articulatory synthesis refers to a model of human articulatory behavior. It is tough to collect the necessary data to construct models and rules.

A text-to-speech synthesizer's digital signal processing component generates the audio waveform that corresponds to the spoken signal, while the natural language processing component provides details about the speech's phonemes and prosody. The vocal tract model is

predicated on the three commonly used techniques of format synthetic, log - likelihood synthesis, & articulatory synthesis. Formant synthesis, which uses formant filters with separate control, is one technique to reproduce the vocal tract's transfer function. As opposed to a noise signal, a stream of impulses is employed as input when making spoken sounds. Despite the difficulties involved in altering the formant synthesizer's settings, they can be determined by adhering to a set of principles generated from an analysis of the characteristics and environment of mobile devices. Like formant synthesis, linear prediction uses a single all-pole filter instead of multiple parallel filters to generate sounds. Before conducting actual study, the linear prediction method leverages available data to create educated assumptions about the voice tract's features. Articulatory synthesis is a method for generating human-sounding speech. It is tough to collect the necessary data to construct models and rules.

All these strategies produce comprehensible, albeit unique, language. The data-driven technique of linear prediction synthesis bolsters a new wave of techniques. The key distinction lies in the fact that the first waveform is still formed using data. These approaches are used to categorize a group of items, each of which possesses a phone type, frequency, and time stamp. The process of voice synthesis is uncomplicated, needing only the selection of the appropriate recorded speech units from a database and their subsequent connection. In these systems, diaphones, which consist of the end of one phone and the beginning of the next phone, are rarely employed as phoneme replacements. Synthesis utilizing Mel-Frequency Cestrum Coefficients (MFCCs), sinusoidal modeling, harmonic noise models, and pitch synchronous overlap and add (PSOLA), and harmonic noise models are all second-generation techniques that can be used to gradually change the pitch and timing of phonemes.

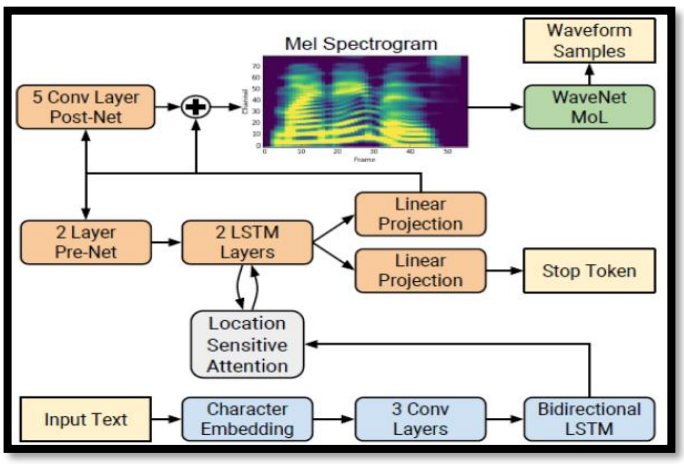

Figure 2.2: Create Real-Time Natural-Sounding Speech from Text by LSTM.

Because all they do is concatenate separate voice tapes for each phoneme, some people refer to second-generation methods as "synthesis via concatenation" techniques. Another well-liked data-driven strategy is inferring the mapping between morpheme definition and important parameters using machine learning and statistical methods. Most of the past research utilized hidden Markov models (HMMs), however other statistical synthesis techniques exist [42]. Modern vocoders commonly use convolutional neural networks to create speech audio waveforms. Google's Wave-Net is an example of a network architecture that uses layers to conduct causal convolutions to achieve cutting-edge performance (for reference, see [43]).

The term "voice conversion" (VC) encompasses a wide range of techniques for masking the origin of a voice. Using the input voice data from one speaker, a VC system can create the sound of a second speaker (the target speaker). In order to develop the model, it is important to have access to a database comprising instances of the voices of both the source and the target speakers. Three elements comprise a VC strategy: analysis, mapping, and synthesis [44]. Aspects of the source and target discourses are extracted during analysis. These qualities must enable routine acquisition of speaker-dependent speech components, alteration of perceptually significant speech aspects, and successful waveform resynthesis [45]. You are free to choose which one to use. Options for TTS synthesis include formants from the vocal tract source filter model, line spectral frequency from linear prediction methods, efficient (MFCCs), and formants from the vocal tract source filter model.

The mapping process converts the distinguishing characteristics of the source speaker's speech into those of the speaker. There are, alas, not many choices. Most used nowadays, the (GMM) is typically trained with retrieved variables and represents a joint distribution between target and source features. In addition, a vector codebook can be improved by considering both sets of properties. The process of frequency warping creates a function that maps the difference between the starting and ending spectra. As soon as the model is trained, new target speech can be matched to untrained source speech.

2.1. RECOGNIZING REAL SPEAKERS FROM AI SYNTHESIZED SPEECH

This article examines cutting-edge strategies for identifying authentic human speech from computer-generated text. We divide these techniques into two groups, each of which helps to distinguish between genuine and fabricated speech. The first group is responsive to strategies

that employ shared traits gleaned from auditory data. This is illustrated through Linear Prediction, Mel-Frequency Analysis, Bi-coherence Spectra, Q- and Cepstral Coefficients (MFCCs), and Other Characteristics such as the Cepstral Coefficient (LPCCs). Bi-spectral analysis and spectrogram-based signal categorization are examined, with an emphasis on the former.

The second category of approaches depends on qualities that can be deduced. These solutions rely on machine learning algorithms that extract model-specific attributes from the input data for the training phase. Experimental results, such as those found in [46], in which convolutional Auto-encoders were used to extract data-driven features, demonstrate that the optimal classification performance is frequently achieved by combining both conventional and data-driven features.

Note that the classification approach described in Chapter 3 employs both traditionally-utilized characteristics, such as mean-kurtosis (MK), and data-driven features created by a recurrent neural network (RNN) (RNN features). We simply merged these two data sets together to construct the mean-kurtosis RNN (MKU) features. MKU features perform better than competing features in both closed-set and open-set voice recognition tasks.

To reiterate, multiple distinct feature sets can be utilized while analyzing voice data (, and to differentiate between real and computer-generated speech). Mel-Frequency Cepstrum Coefficients (MFCCs) are extensively used because they are analogous to the way the human auditory system interprets sound. The spectrum of a signal is converted using a discrete Fourier transform, and a triangle filter bank produces the final product (DFT). A further advantage is that a spectrogram can be quickly created by applying a Fourier transform to the signal. Still widely used in audio analysis, linear inference Smoothing the autoregressive power spectral density of an audio frame can be used to derive the low-pass filtered cepstrum coefficients (LPCCs) [47]. A perceptually-centered temporal-frequency analysis, the constant Q transform yields the constant Q-cepstral coefficients, offer an alternative. In third place is bi-coherence, another classical feature that is part of the group but has not received much attention in speech recognition and processing research.

Our research is driven by the first hypothesis, which employs bi-spectral analysis; in this subsection, we will investigate the second hypothesis, which employs the more typical spectrogram as a distinguishing feature between natural and synthetic speech. The initial step is to implement the standard approach proposed by authors in [4] and [1]. The first of these

investigations begins with a demonstration of how a nonlinear action, such as squaring a sinusoidal input, generates connected harmonics. When the original signal's frequencies and phases are 1 and 2, the square operation generates harmonics with frequencies and phases of 1 + 2, 1 2 and 1 + 2, 1 2. Nonlinear processing produces signal correlations at a higher level that cannot be imposed on the signal and hence cannot be identified by a characteristic such as a power band. Since this is a 3rd-order relationship, it should be recorded in the spectrum.

Comparing the modules and phases of the bi-coherences produced simply comparing the signal before & after a simple square operation shows that the higher level correlations induced by nonlinear processing are clearly visible to the naked eye [4]. Figure 2.3 illustrates the modules and phases of a selection of bi-coherences extracted from both natural and artificial speech. In a second study, classifications based on bi-spectral analysis of voice data are offered ([1]). In addition to traditional human speech, this document contains five distinct forms of cutting-edge synthetic speech.

Using a series of one-versus-all logistic regressions, the authors of [48]–[50] are able to discover these characteristics. Each of these regressions may differentiate between two distinct vocalization groups (either authentic speech or one of the five types of spoofed speech). It is a wonderful method for creating a classification system with numerous levels. Approximately 86% to 86% of spoofed speech samples are correctly categorized such way, while the other samples are incorrectly classified as various types of faked speech. In conclusion, this is an groundbreaking finding that can be used to identify genuine from fabricated speech. This book's subsequent chapters offer a more in-depth examination of the findings. We hope to duplicate them by training the system on a much larger corpus of human voices. We'll also demonstrate several new ways that have emerged from prior research and are now poised to produce even more effective and widespread results. A second option that piques my interest is using a spectrogram of the data to differentiate between authentic and false speech. The complete procedure is detailed in [51]–[53].

The spectrogram and Mel-Frequency spectrogram are extracted from recordings containing both natural human speech and three types of cutting-edge synthesized speech. The spectrogram is constructed first constructing overlapping windows from the input and then applying a short-time Fourier transformation to those windows. A simple transformation of the frequency axis yields the Mel-Frequency spectrogram, which displays a more irregular spacing between spatial frequency samples than the original spectrogram. This transformation uses a psychoacoustic

model of the human ear to account for the fact that different sounds have diverse frequency contents and are perceived at varying intensities.

A convolutional neural network classifier receives input images containing both of these characteristics (CNN). Here, the network has been trained to execute binary classifications, enabling it to discriminate between real and false speech without also identifying imitations based on their manufacturing methods. Similarly, the results shown here are quite encouraging and astoundingly accurate, particularly for Mel-frequency spectrogram classifications.

Data-driven characteristics provide an additional set of relevant features for distinguishing between natural and synthetic speech. This information is frequently extracted using machine learning techniques, and training the chosen model requires a significant amount of voice input data.

Several studies indicate, however, that the effectiveness of classifiers can be significantly improved by combining traditional and data-driven features. However, these changes are often insufficient to deliver performance that exceeds the industry norm. Numerous feature extraction models are available, all of which are based on machine learning. For the purposes of this provision, we will investigate the autoencoder-based technique presented in [54]–[56] for extracting data-driven features from a baseline feature collection. These qualities are used to distinguish between authentic speech and speech generated by various replay systems without prior knowledge of spoofing techniques. Numerous parts were derived from recordings of genuine speech. Aspects of the spectrogram, linear prediction, and Q-cestrum coefficient constancy MFCCs are connected to ideas such as centroids of spectral bands, complex centroids, and low-power-centroid coefficients.

The encoder component of the network generates data-driven features for every traditional feature set, and an auto-encoder is subsequently utilized to augment every feature set after the first feature extraction. The performance of a classifier based just on Gaussian mixture model extensive background model is then assessed using the mean square error (MSE) (GMMUBM) (EER). Three consecutive trials on the qualities used to categorize the items provide distinct outcomes. In contrast to the first experiment, which depended entirely on classical traits, the second experiment, which relied solely on data-driven features, did not produce the same positive outcomes. The third study, which incorporated both traditional and data-driven methodologies, produced the most promising findings, as projected.

3. METHODOLOGY

This chapter focuses on an important issue: how to distinguish between AI-generated speech and human speech. In addition, we provide our proposed strategy for tackling this issue. This technique can be divided into the extraction of attributes and categorization stages.

3.1. PROPOSED METHODOLOGIES

As stated previously, our proposed strategy significantly relies on feature extraction and categorization. Feature extraction must be done in two stages as suggested. Three more feature sets are produced following the computation of bicoherences for both actual and synthetic speech recordings (namely MK, RNN & MKU features). Closed set classifications & open set classifications are the two main types of taxonomies (see Figure 3.1). Both two- and multi-class classification tasks fall under the umbrella of closed set classification challenges, and they can be handled with common machine learning techniques like support vector machines (SVM), logistic regression (LR), as well as random forests (RF). In this article, we discuss our open set classification technique, which may be thought of as the creation of a multiclass denotes that can distinguish between known speech classes and suggest the possibility of unclassified ones (i.e., non-used to train the model). Multiple SVMs are used in this technique.

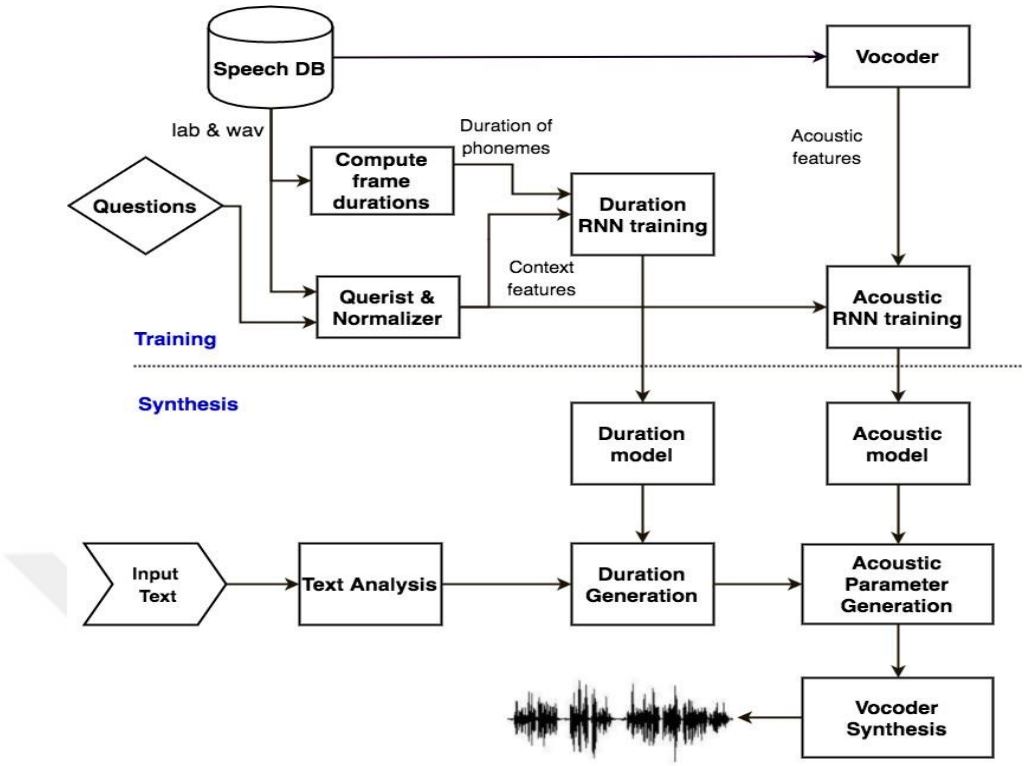

Figure 3.1: Schematic representation of the proposed approach.

3.1.1. Feature Extraction

As seen below, the feature extraction algorithm consists of two phases. Each speech input is first assigned a bi-coherence, and then three sets of features are recovered from the bi-coherences: mean-kurtosis (MK), recurrent neural network (RNN), and mean-kurtosis RNN (MKU). Using these criteria, we will classify the items.

3.1.2. MK Features

based on the synthesis process used to create them, and they have been useful in discriminating between recordings of actual speakers and manufactured recordings [1]. To replicate the conclusions of the authors, we intend to apply their methodology to a new, more diverse group of speech samples.

3.1.3. RNN and MKU Features

Here we show how to use a recurrent neural network (RNN), a specific sort of (CNN), to extract a secondary set of features, or RNN characteristics, from bicoherences. To

summarize Chapter 1's definition of a recurrent neural network (RNN): it is a type of CNN with two modules that act as convolution auto-encoders (encoder & decoder). Its original aim when it was created in [57]–[60] was to segment biological images, but it has since been put to several other tasks, including as anomaly detection and feature extraction. Extraction of RNN features relies on a Recurrent Neural Network (RNN), the architecture of which is shown in Figure 3.3. RNNs use convolutional with subsampling layers in their encoders and convolutional as well as up sampling layers in their decoders. Concatenation levels in the decoding module use skip connections to mix the output of many shallow layers with that from many deep layers. (See Table 3.1). This data is often analyzed with an RNN, even though the bicoherences' modules in this instance are photos. Remember that the encoder module of an RNN receives an input image and generates a compressed version of that image as its output, whereas the decoder module receives the compressed image as its input (i.e., the encoder's output) and generates a reconstructed image that closely resembles the original.

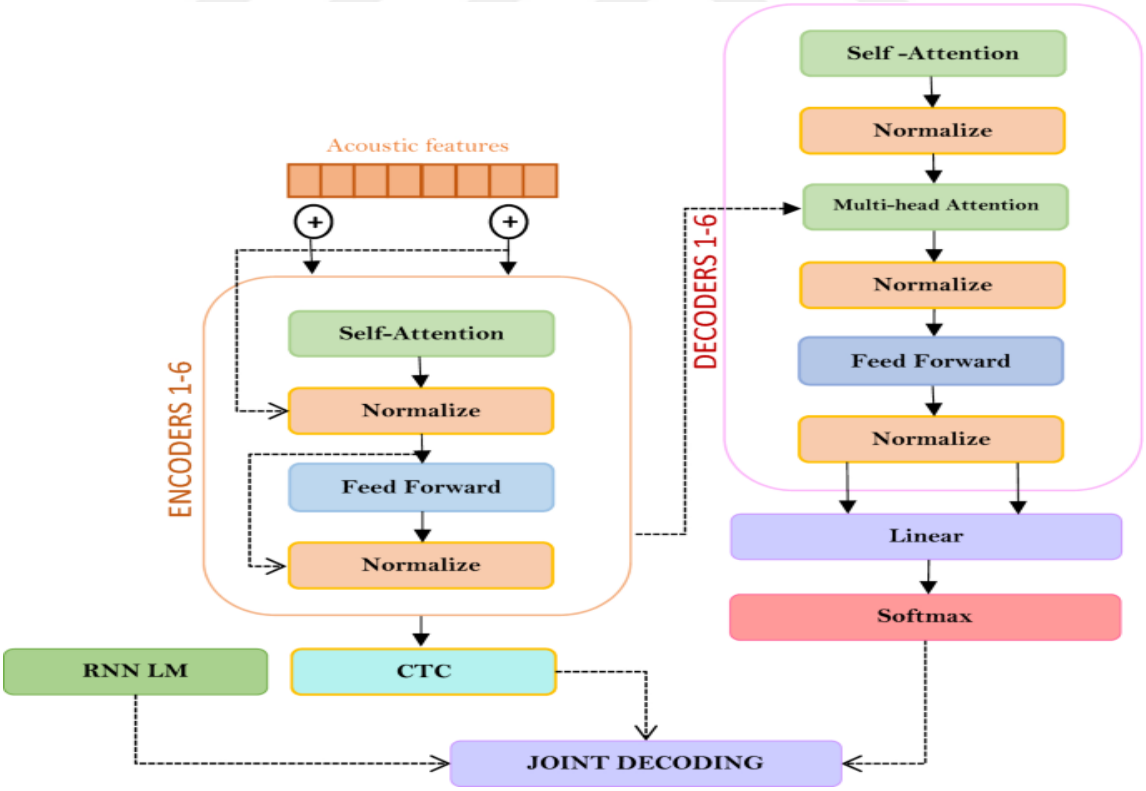

Figure 3.2: Diagram illustrating the RNN's architecture as it is utilized for feature abstraction.

The important impression of a recurrent neural network is that it can reconstruct the set of images from which it was trained with higher precision. Therefore, if we train it with the extracted bi-coherence modules from a given type of speech, say authentic speech, it will be able to rebuild the modules from authentic speech signals more effectively than the modules from spoofed voice recordings. This could result in the development of ways for distinguishing between the two types of speech.

Using the modules of the learnt bi-coherences from actual speech, we will demonstrate how to train an RNN for outlier detection. This technique capitalizes on the RNN's aforementioned properties (and of Auto-encoders in general). Using a variety of RNNs, the modules of the bicoherences estimated across all speech classes are the primary objective for feature extraction. To do this, five distinct RNNs were trained using five distinct synthetic speech categories. The phony accents and accent varieties you hear online are the result of speech synthesis algorithms. We selected these five voice synthesis methods from the available possibilities so that we could generate speech categories that were as distinct to one another as possible. If you would want to learn more about the dataset we utilized, please go to Chapter 4.

3.1.4. Feature Normalization and Dimensionality Reduction

This section describes the normalization processes used to prepare the MK, RNN, and MKU features for classification tasks. In addition, we will describe the approaches for dimensionality reduction that were used to construct the scatter plot.

3.1.5. Feature Normalization

Following extraction, features were normalized to take on values between 0 and 1 so that they could be utilized in machine learning tasks. After normalization, signals x and x'scaled display the mini and max values of signal x. Signal x represents a feature sequence (which will be one of the vectors, created in the preceding sections).

3.1.6. Dimensionality Reduction

We conducted trials employing three distinct dimensionality reduction strategies to classify the features according to the voice synthesis technology that created them. They are useful because they allow us to maintain all vital information while minimizing the number of

elements connected with a voice transmission (say, two or three). There were once only eight MK features and thirteen MKU features (i.e., their discriminating power). In two- or three-dimensional spaces, simplified feature sets can be plotted along axes corresponding to the element values. The figures produced resemble scatter plots. In Chapter 1, we detail the dimensionality reduction techniques we employed, Including, but not limited to, (DSNEA), Independent Components Analysis (pca (ICA), and (PCA) (DSNA). (TSNE). Scatter plots of MK as well as MKU features generated via these approaches are shown in Chapter 4 as an example.

3.1.7. Closed-Set Classification

After collecting and standardizing all the features, it is time to organize them. As previously said, to clarify the methodologies, we will create a clear distinction between closed set classifications and open set classifications. This section focuses on closed set classifications, while the following section does the same for open set classifications. Furthermore, we distinguish between two-class binary classifications (sometimes called closed-set classifications) and multiclass classifications (many sets of categories).

We will refer to the two alternative data categories in a binary closed set classification as "Authentic" and "Fake." The Authentic category often consists of recordings of human voices, whereas the Fake category may contain synthesized speech signals generated by numerous TTS or VC algorithms. A multiclass closed set classification, on the other hand, takes into account more than two categories. One of these is always known as Bonafide, while the others may be known as Alg1, Alg2,..., and AlgN. Within the last two groups, all synthetic speech was generated by a single speech synthesis technique. We also attempt to discern between authentic and fake voice recordings by identifying the precise synthesis technique utilized to generate each bogus signal. Always maintaining separate environments for the training set and test set is a fundamental principle of any machine learning software. according to the sort of quality (MK, RNN and MKU).

3.1.8. Open-Set Classification

Now focusing on the method to open set categorization that employs the MK and MKU features. As previously stated, the objective of an open set classification is to appropriately identify as "unknown" test set instances that correspond to classes the model has never "seen" (i.e., those

non employed in training) (i.e. non employed in the training phase). A benefit of an open set configuration is having more generic outcomes regardless of the dataset used to train the model. Imagine that a new way of voice synthesis has been found, but that only a small number of examples of synthetic speech have been created using this method thus far. Without an update, none of the existing closed set classifiers would be able to determine whether the new instances of speech were fabricated. Even if sufficient data existed to create a training set, this would remain true. The ideal solution to this problem is a classifier that can be trained and evaluated using speech signals from any class or those generated by any speech synthesis technique. In three diverse scenarios, the authors demonstrate the open-set categorization technique. When selecting the first option, you will not utilize the known-unknown attributes. Throughout the training phase of this classification, the participating classes will be designated as Bonafide, Alg1, and Align. Certain traits are utilized as known unknowns throughout the training and testing phases of the other two variations, allowing the training classes to be categorized as Known Unknown, Bonafide, Alg1, Align. Unknown class is a subcategory of synthesized speech that is created during testing of all three versions of the provides flexibility classification approach. This category contains speech synthesized with algorithms other than those employed during the training phase. During model training, "known unknown features" are used to generate a representation of an "unknown" class that resembles the real world but is technically still a mystery. During testing, we only use the genuine unknown characteristics that the model has never encountered before, but these characteristics are unique within that set. We examined all three forms of the open set classifier across two trials, each using a distinct set of authentic speech targets. In the first experiment, authentic speech was used as part of the identified classes, whereas in the second experiment, unlabeled authentic speech signals were employed. This section describes the three variants of our open set classification strategy.

In the first version, five distinct binary classifiers that can differentiate between entities were trained without using the known unknown properties. These five classifiers work together to assign each data point to one of five distinct speech categories. Classifiers were not just tested with data from the five classes used for training (Unknown).

Test examples that were "rejected" (i.e., labeled "the remainder") by all classifiers but one was nonetheless deemed legitimate, and we assigned them to a specific training class. If more than one classifier predicted that an instance belonged to the same training class, we chose the class based on the values of the scores, the outputs of the five classifiers, which also indicate the

probability that an illustration belongs to the training classes (i.e., we delegated the examples to the class corresponding to the classifier's highest score). If the input was labeled as "other" by all classifiers, it was deemed unidentified. As evidenced by the classification findings, most recognized cases were correctly identified, and the majority of recognized speech was assigned to one of the recognized groups, confirming the "pessimistic" nature of the model. We overcome this issue by applying the other two dependent-on-known-unknown-characteristics variations of the open

In the second version of our classification approach, we use six classifiers, five of which are identical to the initial classifiers and one of which must be trained using known-unknown features. To separate the Recognized speech class from of the rest of the training classes, a second classifier has been developed. For clarity, we'll state up upfront that the "remainder" label for the remaining 5 classifiers refers to a combination of known and unknown speech. The model was rigorously tested with both open and hidden data. Features that were approved by one classifier and rejected via the other were once again identified and classified using the score values. If all classifiers failed to correctly identify the characteristics, the situation was labeled as Unknown, and if at least one classifier did, it was labeled as Known-Unknown.

Based on these results, we determined that many genuine unknown features were incorrectly labeled as known-unknown. Leveraging the efficacy of our classification approach, we developed a third variation that is nearly identical to the second save for the fact that it labels as "Unknown" instances that the model categorized as both actual known and recognized. This method allowed us to dramatically enhance the reliability of our data classifications. To train classifiers for all three variants of the open set classification strategy, we used support vector machines, one of several machine learning classification techniques. To elaborate, every study included both well-established and novel elements taken from real-world contexts.

4. RESULTS

In below figures we can see the overall features that we have observed throughout the running of dataset, so perceptively every line shows the single features of the dataset

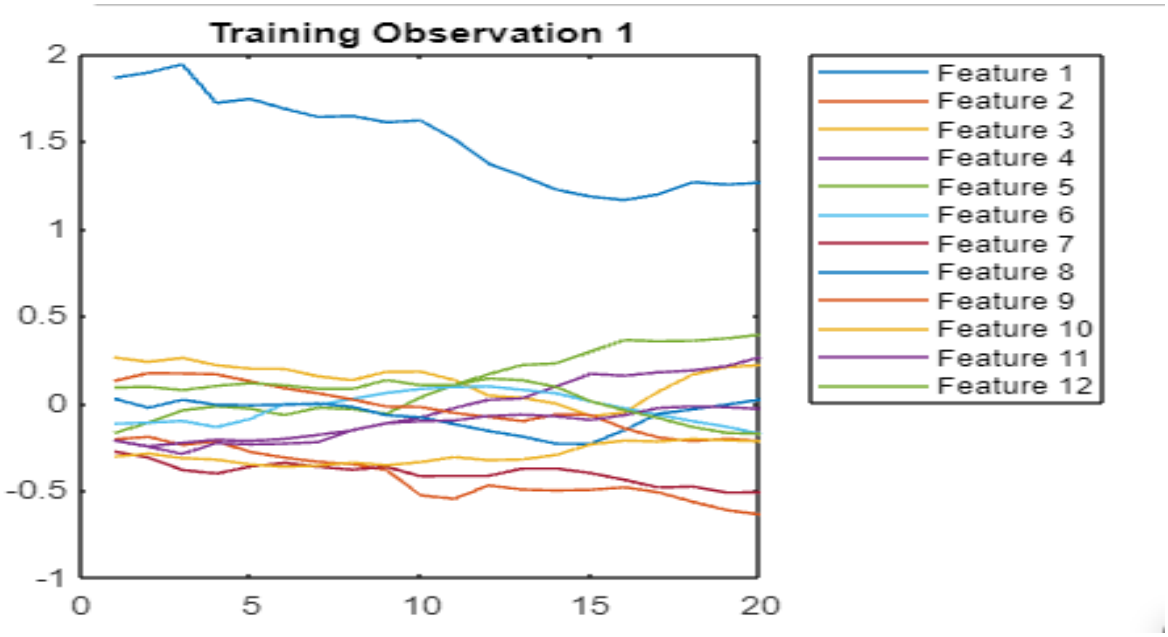

Figure 4.1: Overall features that we have observed throughout the running of dataset

the Japanese Vowels data set after following the instructions from the MATLAB, and this data is avialible in every version of MATLAB, so we do not required any directory setting or anything else, in japanese vowel data set have XTrain and YTrain cell array with 270 sequences of varying lengths and 12 LPC cepstrum coefficients. The category labels for the object Y are 1, 2,..., and 9. XTrain entries are made up of a matrix with 12 rows, one row for each feature, and a number of columns (one column for each time step). , the results below shows the iteration vs accuraty and secondly iteration vs loss, both the data representation in form of graph are shown below

Figure 4.2: Iteration vs accurate and secondly iteration vs loss, both the data representation

MATLAB CODE USING RNN

```
clc
clear all
close all
[ARRAY_1_ON_Y_TRAIN_DATA,ARRAY_2_ON_Y_TRAIN_DATA] =
simpleseries_dataset;
MINIMUM_BATCH_PREDICTION = layrecnet(1:2,10);
[Xs,Xi,Ai,Ts] =
preparets(MINIMUM_BATCH_PREDICTION,ARRAY_1_ON_Y_TRAIN_DATA,ARRAY
_2_ON_Y_TRAIN_DATA);
MINIMUM_BATCH_PREDICTION =
train(MINIMUM_BATCH_PREDICTION,Xs,Ts,Xi,Ai);
```

In below graph we can see the neural network diagram which can be observed after running the MATLAB program , and click on Network diagram, we can see the network diagram with input and hidden layers also output of hidden layer , which can be observed below

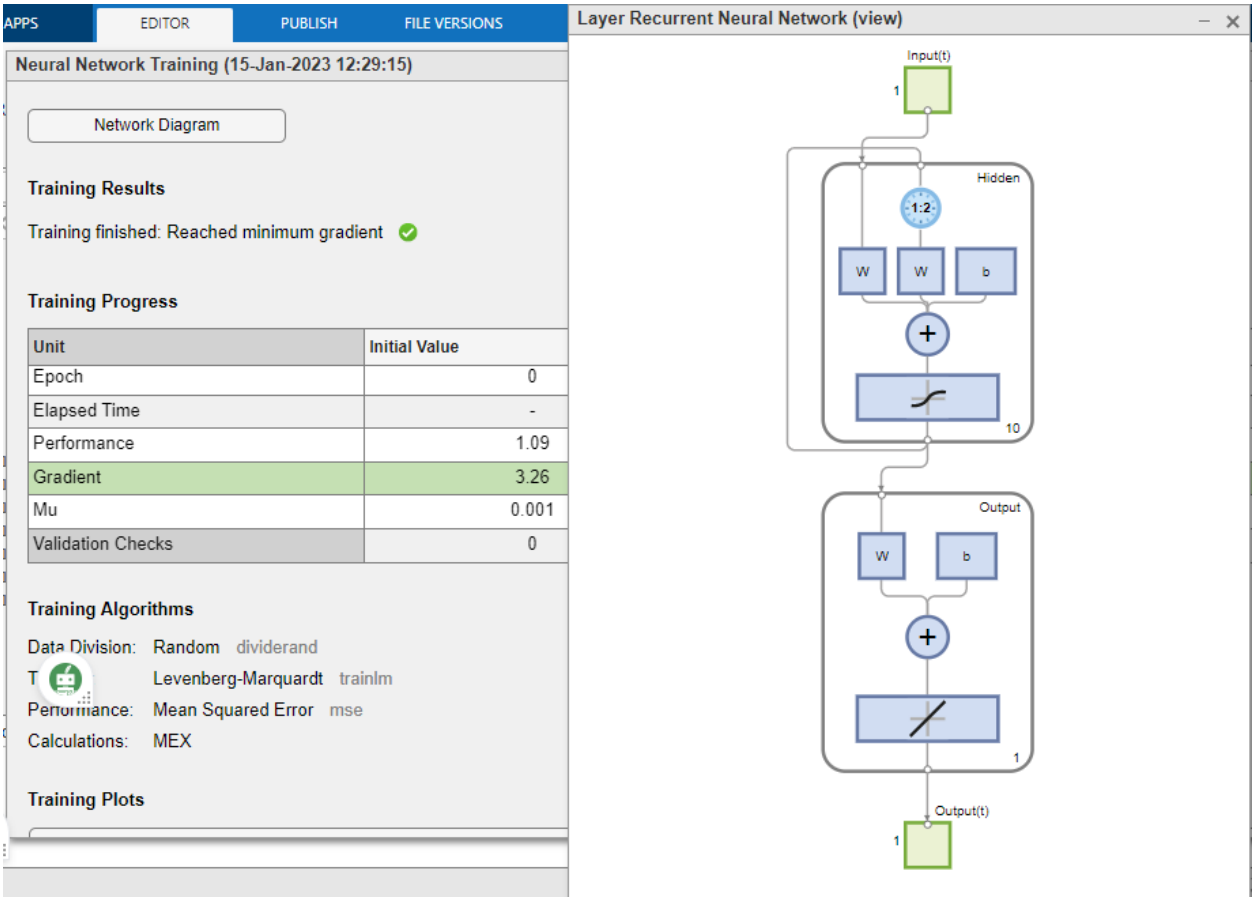

Figure 4.3: The network diagram with input and hidden layers also output of hidden layer

As we run the program we can iteration that goes on with perspective time of fram , so number of epoch out of its maximum can be observed below

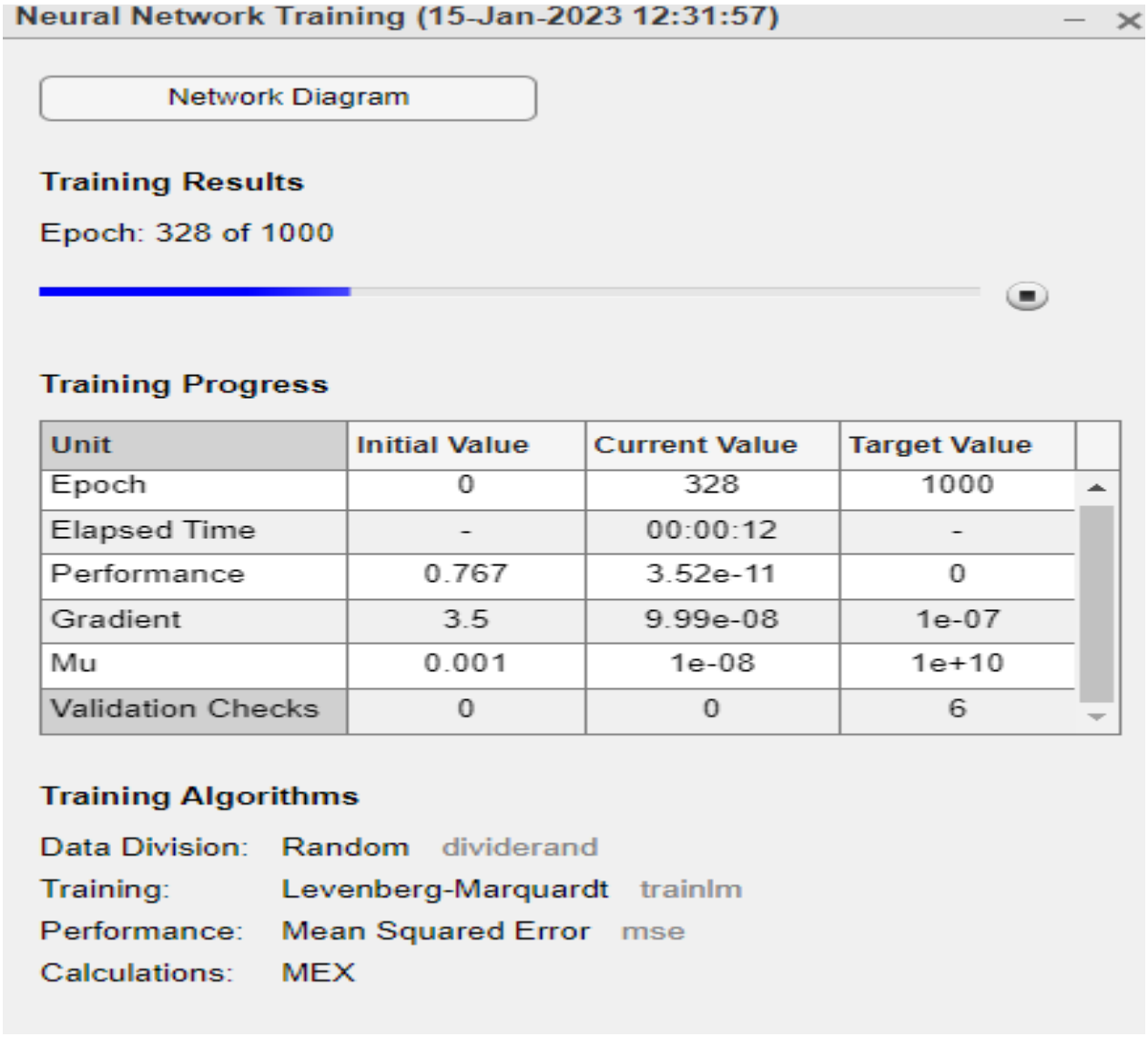

Figure 4.4: Perspective time of frame with number of epoch out of its maximum

In below graph we can see the graph which is exponentially decreasing , as the frame of data changes according to the simulation

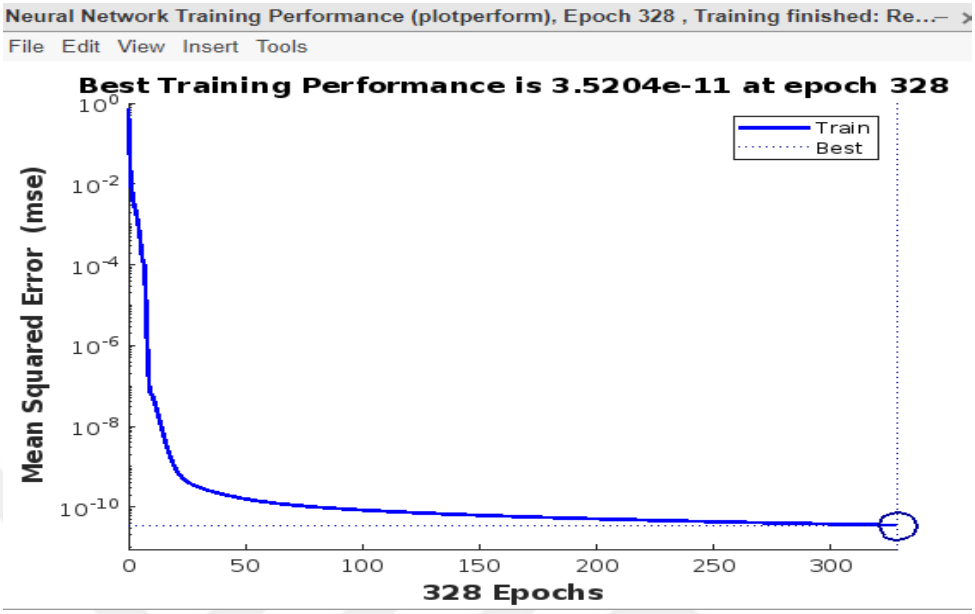

Figure 4.5: Exponentially decreasing , as the frame of data changes according to the simulation

The state for training data can be observed below , having three graphs , so the first one represent the gradient , second one represent the Mu and last one is the validation check which applied on most of the series as per as the time frame

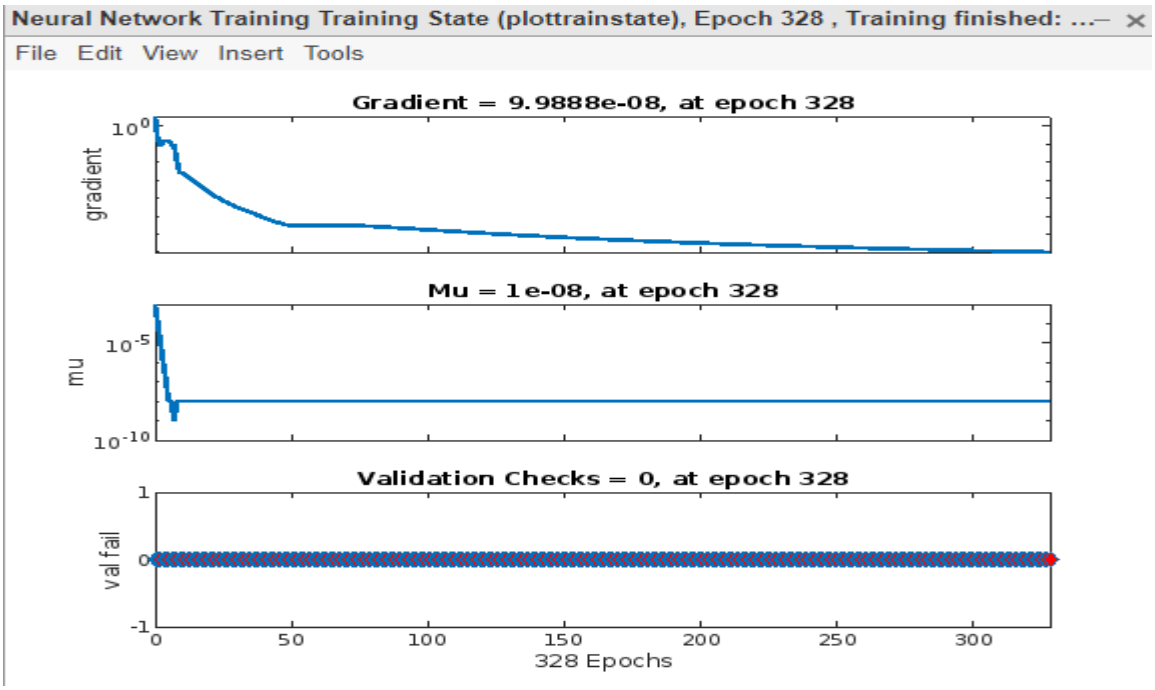

Figure 4.6: The first one represent the gradient , second one represent the Mu and last one is the validation check which applied on most of the series as per as the time frame

So in histogram plot we have observed that the data for the bins with training state and its initial or zero error have been shown in discrete form , where error are at peak when it become equal to 1.02e-07 and total instance it is covered 18 at its high peak

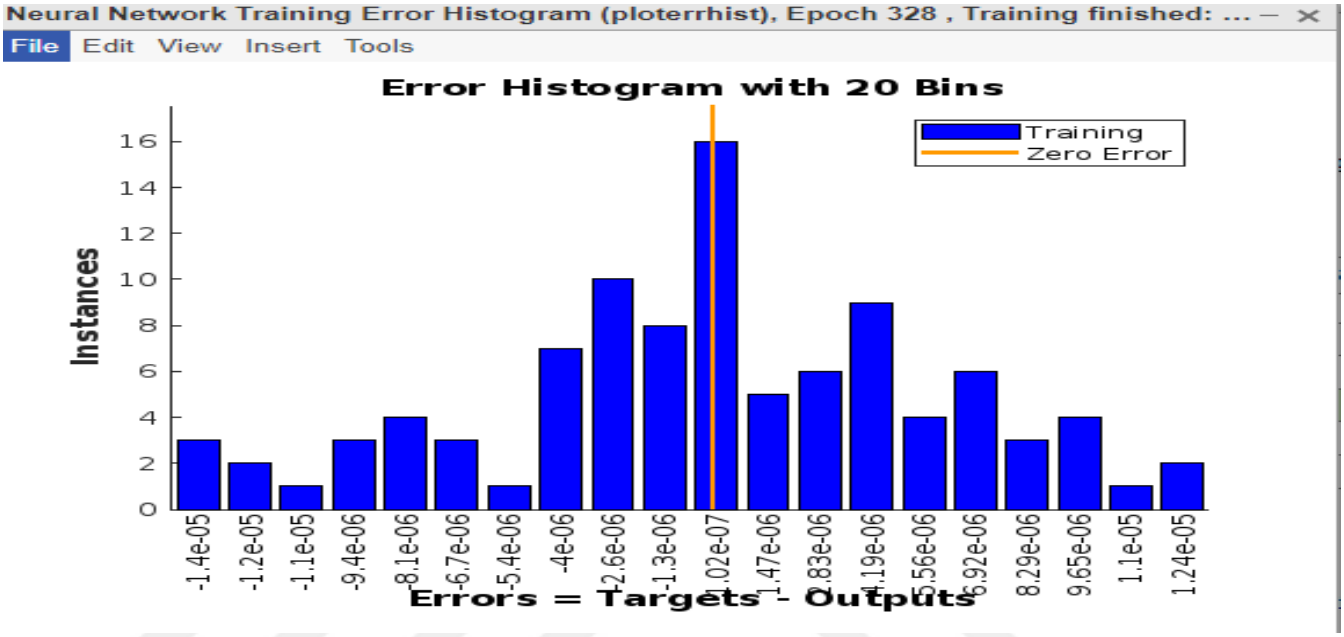

Figure 4.7: Error are at peak when it become equal to 1.02e-07 and total instance it is covered 18 at its high peak

The below is the regression plot for our training data , the regression graph is one line graph have data ovelay on its fit values , where a regression graph and data discreate values shows that system is linear for target as per as its output

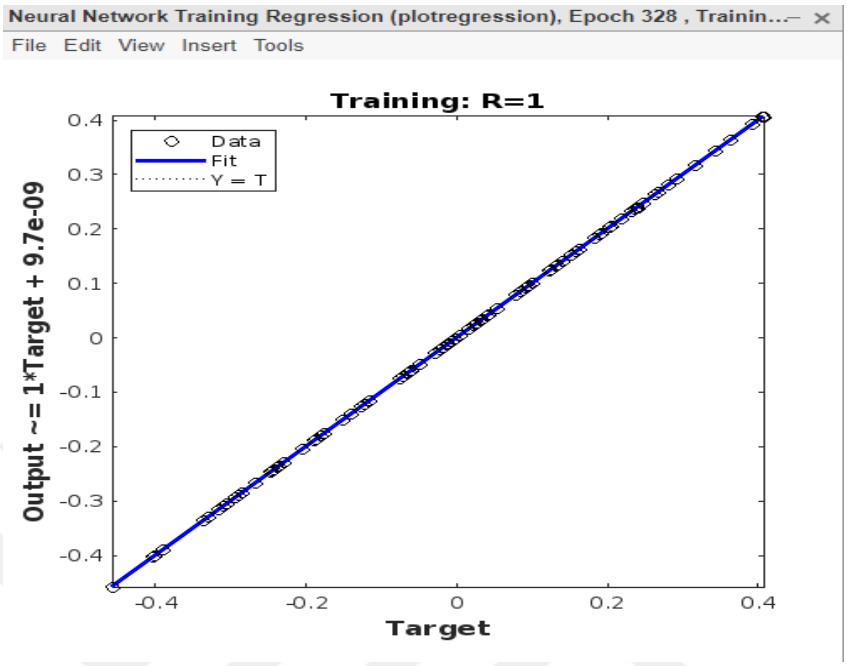

Figure 4.8: Regression graph and data discreate values shows that system is linear for target as per as its output

Also in below figure we have observed time vs error against the length of the data , another graph we have observed the training targets that is achieved , also training output , response of the training data against the time , all the observation can be observed in below graph

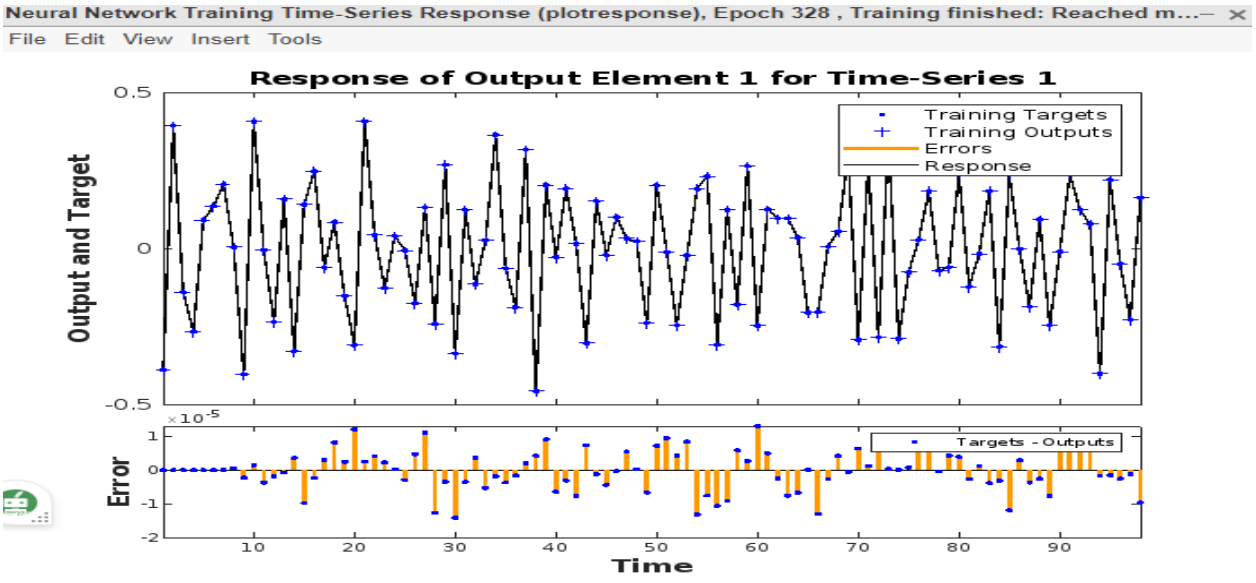

Figure 4.9: Response of the training data against the time, all the observation

Below is a graphic showing the reported confusion matrix and performance evaluation results. The proposed model has an overall accuracy of 99.10%, an F1 score of 85.42%, and a kappa score of 0.87. Similar to this, the model achieved overall accuracy (99.68%), F1 score (84.86%), and kappa score without the age (48) features (0.86).

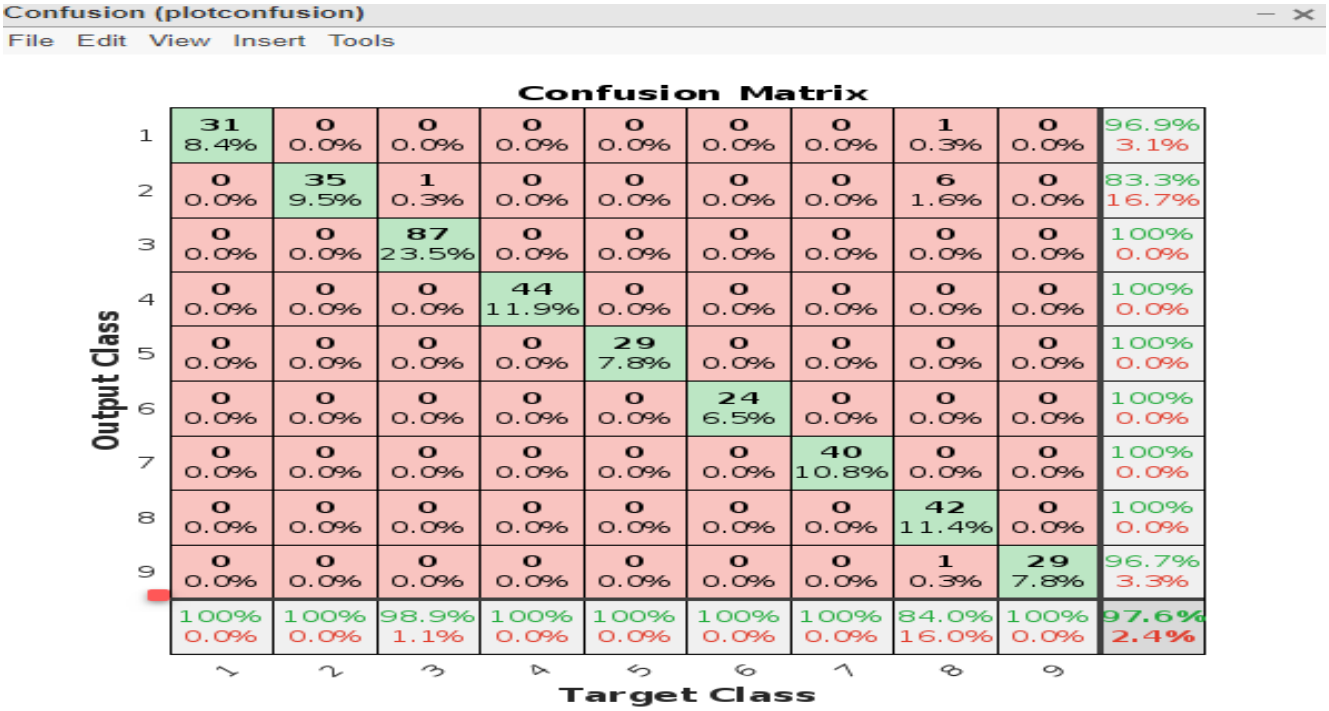

Figure 4.10: The proposed model has an overall accuracy of 99.10%, an F1 score of 85.42%, and a kappa score of 0.87. Like this, the model achieved overall accuracy (99.68%), F1 score (84.86%), and kappa score without the age (48) features (0.86)

By classifying as "unknown" qualities that belonged to the "true unknown" and "known-unknown" classes, our third iteration of our classification approach (OSBT3 and OSST3) yielded the best results for the open set scenario. Since there may be some ambiguity between the unknown and genuine characteristics when the genuine is utilized as the known, the ideal example is one in which the genuine cases are used as the real unknowns (OSST3).

Table 4.1: Comparison of accuracy of existing literature with proposed technique.

Article	Algorithm	Accuracy
[60]	SVM	93.43%
[61]	CNN	98.35%
Proposed	RNN-LTSM	99.76%

In every single feature set (MK, RNN, MKU): We performed three unique confusion matrices (CSM) for multiclass classifications utilizing support vector machines, logistic regressions, and a random forest. We applied the same methods to the binary case (valid versus all the fraudulent classes) and produced three confusion matrices (CSBVAS). The matrices linked with classes with a 100 percent accuracy rate are ignored. At the conclusion, you will find a table containing the accuracy of each closed set classification, including those produced by the convolutional neural network.

5. CONCLUSION

This research falls under the open set classification because it has never been conducted with the purpose of identifying digital forgeries in voice conversations. To date, it appears that classifiers that utilize both known and unknown features throughout the training process may produce the greatest results. Even though these attributes are used to train a model of the "unknown" class, they are nevertheless treated as "unknown" in practice. The ASV-spoof 2019 database is an extraordinarily diversified and diverse collection of speech signals. It includes both natural speech and synthetic speech generated with numerous cutting-edge spoofing techniques using the RNN-LTSM algorithm, which achieves 99.76 percent accuracy. This is just another peculiarity of our job. This makes it excellent for the challenge of differentiating between AI-generated speech and genuine speech. Bi coherence is employed as a speech signal descriptor, but it has not been the focus of many analyses; we believe our findings will help shed light on this topic and motivate further study of sound evidence is called "audio forensics.". Future study may concentrate on understanding a nature of higher order correlations established by speech signal spoofing techniques to develop even more effective descriptors for bi-spectral analysis. Using these notions, we may more accurately characterize the synthesis algorithm used to produce artificial voice signals. Other research, such those that use bi-coherence phase information to extract data-driven attributes or those that simulate an open-ended environment, may also employ convolutional neural networks. The growth of more advanced and accessible forensics tools offers us optimism that we may be able to eliminate the deceptive effects of phony multimedia content from our daily lives in the future. Multiple class label confusion matrices (CSM). These classifications have respective degrees of accuracy of 80 percentage, 64 percentage, and 67 percentage. Forget about confusion conditions for binary classifications; by definition, they are all correct. This thesis finishes by analyzing the relative accuracy of the performance of numerous closed set classifications.

REFERENCES

[1] M. Reynolds, “Courts and lawyers struggle with growing prevalence of deepfakes,” *ABA J (Trial Litig)*, 2020.

[2] J. Kietzmann, A. J. Mills, and K. Plangger, “Deepfakes: perspectives on the future ‘reality’ of advertising and branding,” *Int J Advert*, vol. 40, no. 3, pp. 473–485, 2021.

[3] E. A. AlBadawy, S. Lyu, and H. Farid, “Detecting AI-Synthesized Speech Using Bispectral Analysis.,” in *CVPR workshops*, 2019, pp. 104–109.

[4] D. M. Hawkins, *Identification of outliers*, vol. 11. Springer, 1980.

[5] R. Chalapathy, A. K. Menon, and S. Chawla, “Anomaly detection using one-class neural networks,” *arXiv preprint arXiv:1802.06360*, 2018.

[6] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[7] Y. Zhao, Q.-W. Shen, W. Li, T. Xu, W.-H. Niu, and S.-R. Xu, “Latency aware adaptive video streaming using ensemble deep reinforcement learning,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2647–2651.

[8] V. Nasteski, “An overview of the supervised machine learning methods,” *Horizons. b*, vol. 4, pp. 51–62, 2017.

[9] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, “A systematic review on supervised and unsupervised machine learning algorithms for data science,” *Supervised and unsupervised learning for data science*, pp. 3–21, 2020.

[10] D. Ping Tian, “A review on image feature extraction and representation techniques,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 4, pp. 385–396, 2013.

[11] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[12] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4, no. 4. Springer, 2006.

[13] Y. Cheung and L. Xu, “Independent component ordering in ICA time series analysis,” *Neurocomputing*, vol. 41, no. 1–4, pp. 145–152, 2001.

[14] T. M. Mitchell and T. M. Mitchell, *Machine learning*, vol. 1, no. 9. McGraw-hill New York, 1997.

[15] L. Breiman, “Random forests,” *Mach Learn*, vol. 45, pp. 5–32, 2001.

[16] R. Chalapathy, A. K. Menon, and S. Chawla, “Robust, deep and inductive anomaly detection,” in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10*, 2017, pp. 36–51.

[17] I. J. Cox, M. L. Miller, and J. A. Bloom, “Digital Watermarking, Morgan Kaufmann,” in *2nd National Conference on Innovative Paradigms in Engineering & Technology (NCIPET 2013)–www. ijais. org*, 2001.

[18] M. Barni and F. Bartolini, *Watermarking systems engineering: enabling digital assets security and other applications*. Crc Press, 2004.

[19] I. J. Cox, M. L. Miller, J. Bloom, J. Fridrich, and T. Kalker, “Digital watermarking and steganography morgan kaufmann publishers,” *Amsterdam/Boston*, 2008.

[20] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[21] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 2018.

[22] H. Farid, “Image forgery detection,” *IEEE Signal Process Mag*, vol. 26, no. 2, pp. 16–25, 2009.

[23] A. Piva, “An overview on image forensics,” *Int Sch Res Notices*, vol. 2013, 2013.

[24] L. T. Van, S. Emmanuel, and M. S. Kankanhalli, “Identifying source cell phone using chromatic aberration,” in *2007 IEEE International Conference on Multimedia and Expo*, 2007, pp. 883–886.

[25] K. San Choi, E. Y. Lam, and K. K. Y. Wong, “Automatic source camera identification using the intrinsic lens radial distortion,” *Opt Express*, vol. 14, no. 24, pp. 11551–11565, 2006.

[26] A. E. Dirik, H. T. Sencar, and N. Memon, “Digital single lens reflex camera identification from traces of sensor dust,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 539–552, 2008.

[27] M. K. Johnson and H. Farid, “Exposing digital forgeries through chromatic aberration,” in *Proceedings of the 8th workshop on Multimedia and security*, 2006, pp. 48–55.

[28] I. Yerushalmy and H. Hel-Or, “Digital image forgery detection based on lens and sensor aberration,” *Int J Comput Vis*, vol. 92, pp. 71–91, 2011.

[29] T. Filler, J. Fridrich, and M. Goljan, “Using sensor pattern noise for camera model identification,” in *2008 15th IEEE international conference on image processing*, 2008, pp. 1296–1299.

[30] Z. Fan and R. de Queiroz, “Maximum likelihood estimation of JPEG quantization table in the identification of bitmap compression history,” in *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, 2000, vol. 1, pp. 948–951.

[31] Z. Fan and R. L. de Queiroz, “Identification of bitmap compression history: JPEG detection and quantizer estimation,” *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, 2003.

[32] C. Kraetzer, K. Qian, M. Schott, and J. Dittmann, “A context model for microphone forensics and its application in evaluations,” in *Media Watermarking, Security, and Forensics III*, 2011, vol. 7880, pp. 253–267.

[33] M. Kajstura, A. Trawinska, and J. Hebenstreit, “Application of the electrical network frequency (ENF) criterion: A case of a digital recording,” *Forensic Sci Int*, vol. 155, no. 2–3, pp. 165–171, 2005.

[34] A. J. Cooper, “The electric network frequency (ENF) as an aid to authenticating forensic digital audio recordings–an automated approach,” in *Audio Engineering Society Conference: 33rd International Conference: Audio Forensics-Theory and Practice*, 2008.

[35] H. Malik and H. Farid, “Audio forensics from acoustic reverberation,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 1710–1713.

[36] J. R. Hopgood and P. J. W. Rayner, “Blind single channel deconvolution using nonstationary signal processing,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 476–488, 2003.

[37] H. Guan *et al.*, “MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation,” in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019, pp. 63–72.

[38] A. J. Cooper, “Detecting butt-spliced edits in forensic digital audio recordings,” in *Audio Engineering Society Conference: 39th International Conference: Audio Forensics: Practices and Challenges*, 2010.

[39] H. Farid, “Detecting digital forgeries using bispectral analysis,” 1999.

[40] N. Nitanda, M. Haseyama, and H. Kitajima, “Audio-cut detection and audio-segment classification using fuzzy c-means clustering,” in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, vol. 4, pp. iv–iv.

[41] M. Kyperountas, C. Kotropoulos, and I. Pitas, “Enhanced eigen-audioframes for audiovisual scene change detection,” *IEEE Trans Multimedia*, vol. 9, no. 4, pp. 785–797, 2007.

[42] R. Yang, Z. Qu, and J. Huang, “Detecting digital audio forgeries by checking frame offsets,” in *Proceedings of the 10th ACM Workshop on Multimedia and Security*, 2008, pp. 21–26.

[43] D. P. Nicolalde and J. A. Apolinario, “Evaluating digital audio authenticity with spectral distances and ENF phase change,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1417–1420.

[44] D. Barchiesi and J. Reiss, “Reverse engineering of a mix,” *Journal of the Audio Engineering Society*, vol. 58, no. 7/8, pp. 563–576, 2010.

[45] X. Wang *et al.*, “ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech,” *Comput Speech Lang*, vol. 64, p. 101114, 2020.

[46] T. Dutoit, *An introduction to text-to-speech synthesis*, vol. 3. Springer Science & Business Media, 1997.

[47] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, “Speech synthesis based on hidden Markov models,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[48] H. Kawahara, “STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds,” *Acoust Sci Technol*, vol. 27, no. 6, pp. 349–353, 2006.

[49] M. Abadi *et al.*, “Tensorflow: a system for large-scale machine learning.,” in *Osdi*, 2016, vol. 16, no. 2016, pp. 265–283.

[50] N. Ketkar and E. Santana, *Deep learning with Python*, vol. 1. Springer, 2017.

[51] P. Taylor, *Text-to-speech synthesis*. Cambridge university press, 2009.

[52] Y. Agiomyrgiannakis, “Vocaine the vocoder and applications in speech synthesis,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015, pp. 4230–4234.

[53] A. Sharma *et al.*, “Fast Griffin Lim based waveform generation strategy for text-to-speech synthesis,” *Multimed Tools Appl*, vol. 79, pp. 30205–30233, 2020.

[54] A. van den Oord *et al.*, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.

[55] N. V. Aguerre González, “Neurocognitive Mechanisms of Mindfulness and Grit Traits: An Individual Differences Approach to Cognitive Control,” 2021.

[56] B. T. Balamurali, K. E. Lin, S. Lui, J.-M. Chen, and D. Herremans, “Toward robust audio spoofing detection: A detailed comparison of traditional and learned features,” *IEEE Access*, vol. 7, pp. 84229–84241, 2019.

[57] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 2015, pp. 234–241.

[58] P. R. M. Júnior, L. Bondi, P. Bestagini, S. Tubaro, and A. Rocha, “An in-depth study on open-set camera model identification,” *IEEE Access*, vol. 7, pp. 180713–180726, 2019.

[59] M. Hassen and P. K. Chan, “Learning a neural-network-based representation for open set recognition,” in *Proceedings of the 2020 SIAM International Conference on Data Mining*, 2020, pp. 154–162.

[60] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[61] M. Morise, F. Yokomori, and K. Ozawa, “WORLD: a vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE Trans Inf Syst*, vol. 99, no. 7, pp. 1877–1884, 2016.